Delay line
By Anton H.C. Smith

Further discussions with John Anderson and Mike Utes, there seems to be a much simpler solution to the Delay line problem. It seems it is possible to create the delays needed by delaying the source of the control signals. This means delaying the Sequencer, more specifically the NRZ signal going to the Sequencer.

The NRZ signal is a basic data transmission protocol. NRZ consists of a continuous stream of seven-bit packets, each bit being the same duration as one Tevatron RF cycle. Four bit of this seven bits packet contains the commands necessary to create all the control signal that Sequencer create to drive the AFE, more information about the NRZ signal can be found on the following link, http://d0server1.fnal.gov/users/utes/webpage/svxfiles/svxseqdoc_final.pdf. Thus delaying this NRZ will cause a delay in all the control signals. The advantage of creating the delay this way is as follows:

1) Only the NRZ signal is needed to be delayed
2) All considerations that were needed for delaying signals individually no longer apply.

In the discussion with Mike Utes, two method of creating the delay of the NRZ was discussed these methods are:

1) Creating the delays on the Sequencer board
2) Creating the Delays in the Sequencer Controller

Creating the delays on the Sequencer board

It is possible to delay the NRZ signal as it enter the Sequencer. Fortunately, in design the Sequencer board, Mike Utes create space and wiring connection to introduce a spare PLD. Thus to create the delay all one has to do is re-route the NRZ signal through this newly introduce PLD, which in turn will create the necessary delay and return the delayed NRZ signal to the appropriate sections of the sequencer.

Creating the Delays in the Sequencer Controller

It is possible to delay the generation of the NRZ signal. Fortunately, in the design of the Sequencer controller there are programmable delay lines. Thus it is possible to program in the appropriate delays.

Delay in Sequencer vs Delay in Sequencer Controller

Delay in Sequencer	Delay in Sequencer Controller
Hard-ware modification needed	No Hard-ware modification needed
No extra rack space	Needs its own crate
Delays cannot be remotely change	Delays can be change remotely.
Code need to written for PLD	No new code

Consideration for delay

There is a finite resolution of the delay when considering using the Sequencer Controller delay method. This because in the Sequencer Controller the programmable delay is in blocks of;

1) 9x132 ns
2) 7x18.8ns
3) 9x2.5ns

The first two delay blocks are D-Flip Flops and the third are analog delay lines.

The other consideration is one that is inherent in any method to provide the delay. This is the dead time of the system during Readout. To prevent this dead time it means that the Trigger and the Readout process should not be delayed. An obvious question is whether the seemly small delays that is be implemented would affect the acquisition of data.

If the dead time cannot be tolerated then a system of logic would need to prevent the dead time in delay using the Sequencer. This logic is already provided in the Sequencer controller method.

Conclusion

Delaying by way of the Sequencer involves writing code for the PLD, modifying existing hardware and it does not lends itself for the delay to be control remotely. Delaying using the Sequencer Controller requires no hardware modification or new software to be written it also lends itself for the delay to be control remotely but it does call for a new crate. This is because the Sequencer control would be specific for our use only. Also the resolution of the delay is limited by the existing allowable programmable delay residing on the board.